Simultaneous Cooperative Exploration and Networking Based on Voronoi Diagrams

Jonghoek Kim, Fumin Zhang, and Magnus Egerstedt *

* *School of Electrical and Computer Engineering, Georgia Institute of Technology, USA (e-mail: jkim37@gatech.edu, fumin, magnus@ece.gatech.edu)*

Abstract: We develop a strategy that enables multiple intelligent vehicles to cooperatively explore complex and dangerous territories. Every vehicle drops communication devices and expands an information network while constructing a topological map based on the Voronoi diagram. As the information network weaved by each vehicle grows, intersections eventually happen so that the networks are shared. This allows for distributed vehicles to share information with other vehicles that have also dropped communication devices. Our exploration algorithms are provably complete under mild technical assumptions. A performance analysis of the algorithms shows that in a bounded workspace, the time spent to complete the exploration decreases in proportion to the number of vehicles employed. The algorithms are demonstrated in simulation.

Keywords: Cooperative exploration, Voronoi diagram, Sensor network, Information network

1. INTRODUCTION

Exploration of complex and dangerous territories posts great challenges for robotics research. For mobile robotics, the recent developments in simultaneous localization and map making (SLAM) (Thurn and Burgard (2005),Durrant-Whyte and Bailey (2006),Bailey and Durrant-Whyte (2006)) have provided valuable techniques to answer some of the challenges. Intuition suggests that using cooperative multiple vehicles will increase time efficiency. Coordination of multiple vehicles typically relies on communication between vehicles, but direct communication is easily blocked or at least attenuated by obstacles. Hence one major challenge for a successful multi-vehicle strategy is the lack of line of communication (LOC).

Recently, small, low-cost devices equipped with short range communication and low power sensors, such as the Berkeley MOTES, become commercially available. A large number of such devices may form a sensor network and establish the LOC between each other Culler et al. (2004). In an ideal situation, each device serves as an information node and together they form an information network that relays information between devices.

These developments have inspired us to investigate a strategy named as Simultaneous Cooperative Exploration and NeTworking (SCENT). In this paper, we present SCENT algorithms that construct Voronoi diagrams as topological maps of the workspace. The workspace is considered completely explored if the reduced generalized Voronoi diagram (RGVG), as defined by Choset and Burdick (2000), is fully constructed. The vehicles are initially deployed at arbitrary locations and are not necessarily aware of the existence of other vehicles. Each vehicle starts with a single vehicle strategy developed in our recent work Kim et al. (2009) to explore its surroundings and to construct the information network.

Every vehicle expands the information network while dropping communication devices. As the information network weaved by each vehicle grows, intersections eventually happen so that the networks are shared. This allows for distributed vehicles to share information with other vehicles that have also dropped communication devices. This shared network is a key feature that distinguishes the SCENT approach from other papers that only allow robot-to-robot communication. Notice that we do not have to consider the localization which is a fundamental problem in robotics, since shared network can localize every vehicle at a correct intersection.

Each vehicle expands the information network not overlapping the information networks built by other vehicles. However, there may be the case where the expansion of information network is blocked by the information networks built by other vehicles. These blocked vehicles are then redirected to unexplored regions of the workspace, since shared network allows for distributed vehicles to detect unexplored region. Moreover, shared network is applied to reduce the chance of occurrence of blocking events for multiple vehicles.

Voronoi diagrams have been widely used for topological maps in robotics, c.f. Choset et al. (1996) Lavalle (2006), as well as for studying coverage problems in sensor networks Cortés et al. (2004) Martinez et al. (2007). This paper provides provably complete algorithms in constructing the RGVG. A performance analysis of the algorithms shows that in a bounded workspace, the time spent to complete the exploration decreases in proportion to the number of vehicles employed.

The paper is organized as follows : Section 2 introduces the background knowledge regarding Voronoi diagrams for the workspace of interest, and then reviews exploration algorithms and results for a single vehicle. Section 3 presents cooperative exploration algorithms using multiple vehicles and the construction of the information network. Section 4 analyzes the efficiency of the cooperative exploration algorithms. Section 5 demonstrates simulation results, and section 6 provides conclusions.

2. ALGORITHMS FOR A SINGLE VEHICLE

In this section, we review the exploration algorithms in Kim et al. (2009) that construct the RGVG of the workspace using a single vehicle. We omit some technical details and convergence proofs in this paper.

2.1 Definitions and Assumptions

Consider a connected and compact workspace $W \subset R^2$ with boundary ∂W as a regular curve. Let O_1,O_2,...O_{M-1} be $M-1$ disjoint, compact, and connected obstacles such that $O_i \subset W$. O_M is a "virtual" obstacle that bounds the workspace, i.e., $\partial W \subset \partial O_M$. We denote the set of obstacles S_O by $S_O = \{O_1, O_2, ...O_M\}$.

Obeying the conventions established in the literature on Voronoi diagrams (Aurenhammer (1991); Nagatani and Choset (1999); Lavalle (2006); Klein (1990); Choset and Burdick (2000)), we define the *Voronoi cell* for an obstacle O_i as the set of points that are closer to O_i than to any other obstacle in S_o for $i = 1, 2, ..., M$. $\partial V(O_i)$ is the boundary of the Voronoi cell for O_i, i.e., $V(O_i)$. The *Voronoi diagram* of the workspace is defined as the union of all cell boundaries (Klein (1990)). A Voronoi edge between two Voronoi Cells $V(O_i)$ and $V(O_j)$ is $E_{ij} = \partial V(O_i) \bigcap \partial V(O_j)$.

We define an *intersection* as the point in the workspace W where a circle centered at the point is tangential to obstacle boundaries at more than two points as illustrated in Fig.1. The circle is called an *intersection circle*. Suppose that the vehicle is at an intersection, then the points of tangency on the obstacle boundaries correspond to the closest points, i.e. points that have local minimal distances to the vehicle. The lines connecting the intersection and the closest points on the obstacle boundary partition the intersection circle into *sector*s. We can see that each sector is the "pie shaped area" within the intersection circle as seen in Fig. 1.

The vehicle under control moves along the E_{ij} until it visits an intersection P as depicted in Fig.1. It will detect two closest points on ∂O_i and ∂O_j, since $P \in E_{ij}$. The sector that has these two closest points as its end points is defined as *sector* 0 for the intersection P, as illustrated in Fig.1. Suppose that there are n sectors in the intersection circle as seen on Fig.1. Looking into the page, we then index the sectors in the counter clockwise direction from sector 0. The index k satisfies $0 \leq k \leq n-1$. When two end points of a particular sector are on the same obstacle, the sector is called a *blocked sector* that is illustrated as "sector 2". An *open sector* denotes a sector that is neither a blocked sector nor a sector 0, illustrated as "sector 1" and "sector 3" in Fig. 1.

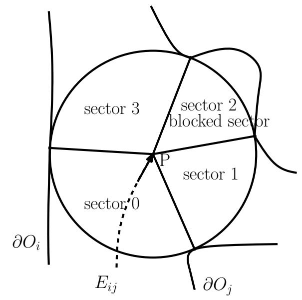

Fig. 1. The position of a vehicle is at the intersection. The *sector i* is the sector adjacent to the sector $i-1$ in the counter clockwise direction.

If the intersection detected by a vehicle has an open sector that has not been visited by the vehicle, then the intersection is marked as *unexplored*. Otherwise, the intersection is marked as *explored*.

The following assumptions are made about the workspace and the vehicle's sensing and localization capability.

(A1) $\partial V(O_i)$ is a simple closed curve for all $O_i \in S_O$. In other words, $\partial V(O_i)$ is continuous and no self-intersection occurs.
(A2) there are finitely many intersections in W. All blocked sectors for these intersections are detectable by the vehicle.
(A3) $\bigcup_{O_i \in S_O} \overline{V}(O_i) = W$.
(A4) the initial position of a vehicle is such that an obstacle other than O_M is detected to the right of the vehicle [1]. The vehicle can distinguish O_M from other obstacles.

2.2 Expanding the enclosing boundary

We call a closed loop that contains intersections connected by Voronoi edges *an enclosing boundary*. The area inside the enclosing boundary is called the *enclosure*.

Our exploration algorithms first construct an enclosing boundary that contains only one obstacle, then the enclosing boundary is expanded by adding one new obstacle at a time. At any moment in our exploration algorithms, the enclosing boundary is unique.

More specifically, the initial enclosing boundary is denoted as B_0. Then we update B_0 to obtain B_k for $k = 1, 2, ...$ until B_k encloses all the obstacles except for O_M. There are $k+1$ obstacles inside B_k where $0 \leq k \leq M-2$.

We expand B_k while maintaining it as a simple closed curve tracked by the vehicle in the clockwise direction. This expansion is performed by first moving through an open sector of an intersection on B_k to construct a candidate segment formed by Voronoi edges, and then to replace certain segment of B_k with the candidate segment.

A set of rules are designed in Kim et al. (2009) to expand the enclosing boundary. We have proved in Kim et al. (2009) that the algorithms finish in finite time and a complete Voronoi diagram is obtained as a result.

[1] Assumption (A4) is strictly speaking not a restriction, since the vehicle can initialize the heading orientation so that an obstacle other than O_M is detected to the right of the vehicle. In the case where multiple vehicles are involved, assumption (A4) is applied to every vehicle.

3. SCENT ALGORITHMS

In this section, we present SCENT algorithms by extending the algorithms for a single vehicle to multiple vehicles. We denote a vehicle as v^i where $1 \leq i \leq N_v$ and N_v is the number of vehicles. Every vehicle v^i deploys communication devices on intersections. If necessary, communication devices are deployed on long Voronoi edges in order to relay data from one intersection to another intersection that is out of maximum radio range. These communication devices then form an information network. The methods for communication devices deployment by a robot platform are research topics of interests that are not the focus of this paper. We also assume the information network is in place once the communication devices are deployed.

Let B^i denote the enclosing boundary built by v^i. Since multiple vehicles are involved, one major modification over the single vehicle exploration algorithm is to expand B^i in such a way that the enclosure inside B^i does not overlap with enclosure inside B^j where $j \neq i$. Therefore, when v^i visits an intersection on $B^i \bigcap B^j$, v^i obeys a rule called *sector selection rule*.

Before stating the sector selection rule, we introduce *pointer* + 1 sector of B^i, which denotes a sector whose index is bigger than the *pointer* sector of B^i by one. Fig.2 gives an illustration for *pointer* sector, *pointer* + 1 sector, and sector 0 stored at every intersection on B^i.

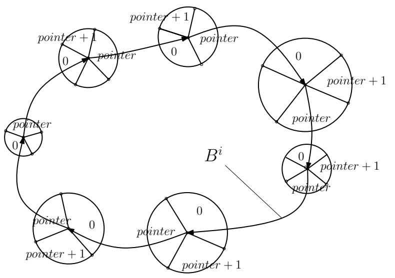

Fig. 2. The illustrative case to show *pointer* sector, *pointer* + 1 sector and sector 0 stored at every intersection on B^i.

When v^i visits an intersection on $B^i \bigcap B^j$, decision for heading direction is performed according to the following sector selection rule :

R1 When the vehicle v^i visits an intersection P on $B^i \bigcap B^j$, the vehicle searches for an open sector in the counter clockwise direction from the *pointer* + 1 sector of B^i to the sector 0 of B^j. Once an open sector is detected, then the following condition is checked. If the vehicle would move through the open sector, then O_M would not lie to the right of the vehicle. If an open sector is detected that satisfies this condition, then the vehicle moves through the open sector. Otherwise, the vehicle moves through the *pointer* sector of B^i at P.

Note that, using the sector selection rule, the vehicle v^i does not move into the enclosure of B^j where $j \neq i$.

Our results in this section require some basic knowledge of graph theory (Ferrari-Trecate et al. (2006)). An undirected graph G is defined by a set $N(G)$ of nodes and a set $E(G) \subset N(G) \times N(G)$ of edges. Two nodes x and y are *neighbors* if $(x, y) \in E_G$. A graph G is *connected* if there is a path connecting every pair of distinct nodes. The *subgraph* G_s of G is the pair $(N(G_s), E(G_s))$ where $N(G_s) \subset N(G)$ and $E(G_s) = \{(x, y) \in E(G) : x \in N(G_s), y \in N(G_s)\}$. We can write $G_s \subset G$.

3.1 Communication Graph

We define a *communication graph* as the graph where every node represents a deployed communication device and every edge represents a communication link. The nodes and edges of the communication graph are time-varying, since new node and edges are added to the graph when a vehicle deploys a communication device. For each vehicle, we distinguish three subgraphs, $G^i(t)$,$\hat{B}^i(t)$, and $C^i(t)$.

$G^i(t)$ where $i = 1, 2, ..., N_v$ is the communication subgraph where every node represents a communication device deployed by vehicle v^i. Since B^i is used to represent the enclosing boundary built by v^i, we use the notation $\hat{B}^i(t)$ for the communication subgraph where all nodes are on the enclosing boundary B^i. $N(\hat{B}^i(t))$ is the set of nodes along B^i, and $E(\hat{B}^i(t))$ is the set of edges of $\hat{B}^i(t)$. A special case here is that we allow $i = 0$ so that $\hat{B}^0(t)$ is the subgraph where all nodes are on $\partial V(O_M)$.

Suppose that a communication device is already deployed by v^i at an intersection and that v^j visits the intersection. Then, through the communication device deployed at the intersection, v^i can relay data structure of $G^i(t)$ and $\hat{B}^i(t)$ to v^j and vice versa. Note that v^j does not have to drop a communication device at the intersection, since communication link is already established through the communication device deployed by v^i.

Hence, both vehicles are aware of the structure of the combined communication graph. In this way, each vehicle v^i builds a combined communication graph $C^i(t) = (N(C^i(t)), E(C^i(t)))$ that is the maximally connected graph such that $G^i(t) \subset C^i(t)$ where $i = 1, 2, ..., N_v$. The relationship among the communication subgraphs is that $\hat{B}^i(t) \subset G^i(t) \subset C^i(t)$. Every vehicle v^i stores one $C^i(t)$ and all $\hat{B}^j(t)$ where index j is determined such that $G^j(t) \subset C^i(t)$.

The vehicle v^i uses $C^i(t)$ to find an unexplored intersection for building a new enclosing boundary at a new position. This is explained in the next subsection.

3.2 Resolve blocking or overlapping events

Every vehicle v^i expands B^i in a way that the enclosure of B^i does not overlap with the enclosure of B^j where $j \neq i$. However, there may be the case where the expanding enclosing boundary B^k is blocked by the enclosing boundaries constructed by other vehicles as illustrated on Fig.3. *Blocking* of B^k denotes the situation when $E(\hat{B}^k(t)) \subset \bigcup_{n \neq k} E(\hat{B}^n(t))$.

Another situation that is associated with blocking is overlapping. This can happen if the initial enclosing boundaries built by two different vehicles are identical. This is possible at the beginning of the exploration, if the initial position

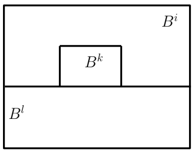

Fig. 3. The expanding enclosing boundary B^k is blocked by the enclosing boundaries constructed by other vehicles.

of v^i is that the obstacle to the right of v^i is also to the right of v^j. If $N(\hat{B}^i(t)) \subset N(\hat{B}^j(t))(j \neq i)$ when B_0^i is built, then we define this case as *overlapping* of B^i.

In the case where blocking or overlapping occurs, we redistribute the vehicles by directing the blocked or overlapped vehicle to an unexplored intersection where a new enclosing boundary can be built. As long as there is an unvisited Voronoi edge in W, unexplored intersection exists on $C^i(t)$ for every vehicle v^i. This is stated as Lemma 1.

Lemma 1. If there exist an unvisited Voronoi edge in W, then there exists an unexplored intersection on $C^i(t)$ for all i.

Proof. We prove by contradiction. Suppose that all the intersections on $C^i(t)$ are explored. This implies that all the Voronoi edges connected to $C^i(t)$ are already visited by vehicles, and communication devices are deployed along the edges. Thus, the edge set of $C^i(t)$ does not contain any edges that lead to unexplored regions. This can only be true if $C^i(t)$ has all the Voronoi edges in W, which implies that all the Voronoi edges in W have been visited by vehicles. This is a contradiction. □

The redirecting strategy works as follows. When blocking or overlapping occurs, then v^i searches for an unexplored intersection on $C^i(t)$. Note that this unexplored intersection will not lie on a blocked enclosing boundary. This is stated as the following Lemma 2.

Lemma 2. If an unexplored intersection is found on $C^i(t)$, this unexplored intersection is not on a blocked enclosing boundary.

Proof. We prove by contradiction. Suppose that an unexplored intersection is found on a blocked enclosing boundary B^j. This implies that there exists an unvisited edge that intersects the unexplored intersection. This unvisited edge can not be inside B^j, since all the Voronoi edges inside B^j are visited by v^j (Theorem 2 in Kim et al. (2009)). Since enclosing boundary B^j is blocked, we have $B^j \subset \bigcup_{n\neq j} B^n \bigcup \partial V(O_M)$. Then there exists B^k such that the unvisited edge is inside B^k where $B^j \bigcap B^k \neq \emptyset$. However, all the Voronoi edges inside B^k are visited by v^k (Theorem 2 in Kim et al. (2009)). Hence, the unvisited edge can not be inside B^k either. Therefore, unexplored intersection can not exist on a blocked enclosing boundary. □

By applying the breadth-first search algorithm on $C^i(t)$, v^i can find the shortest (hop distance) path from the current position of v^i to all the unexplored intersections on $C^i(t)$. Among these unexplored intersections, v^i selects the one with the smallest hop distance and marks it as Q_{v^i}. The position of Q_{v^i} is relayed (broadcasted) across $C^i(t)$ to all other vehicles sharing $C^i(t)$. In the case where v^j visits Q_{v^i}, v^j ignores Q_{v^i} without changing B^j. In this way, Q_{v^i} is "reserved" for v^i until it is reached by v^i. Once v^i reaches Q_{v^i}, it builds a new enclosing boundary.

Suppose that blocking or overlapping event occurs for v^j while v^i is moving toward Q_{v^i}. In this case, by applying the breadth-first search algorithm on $C^j(t)$, v^j can find the farthest (hop distance) unexplored intersection from Q_{v^i}. Then v^j marks it as Q_{v^j} followed by moving along the shortest path from the current position of v^j until it reaches Q_{v^j}. Q_{v^j} is relayed (broadcasted) across $C^j(t)$ to all other vehicles sharing $C^j(t)$.

This strategy relies on the availability of at least one unexplored intersection for each blocked or overlapped vehicle. Hence, we make the following assumption :

(A5) When blocking or overlapping event occurs for v^i, there exists at least one unexplored intersection, except for Q_{v^j} $(j \neq i,\ G^j(t) \subset C^i(t))$, on $C^i(t)$.

This assumption seems too strong. It is possible that blocked or overlapped vehicle v^i can not find an unexplored intersection, except for Q_{v^j} $(j \neq i,\ G^j(t) \subset C^i(t))$, on $C^i(t)$. Hence, we develop a rule, which is introduced in the next subsection, so that we can reduce the chance of occurrence of blocking events for multiple vehicles.

3.3 Avoid blocking using the communication graph

Every vehicle v^i obeys the following *blocking avoiding rule* which is to avoid blocking events of $\hat{B}^k(t)$ where index k is determined such that $k \neq i$ and that $\hat{B}^k(t) \subset C^i(t)$.

- If expansion of B_n^i, which denotes the enclosing boundary for v^i updated after n steps, leads to $E(\hat{B}^k(t)) \subset \bigcup_{m\neq k} E(\hat{B}^m(t))$ where $k \neq i$ and $\hat{B}^k(t), \hat{B}^m(t) \subset C^i(t)$, then the expansion will not be performed.

This blocking avoiding rule is to avoid blocking of B^k where $k \neq i$ and $\hat{B}^k(t) \subset C^i(t)$. Note that if expansion of B_n^i leads to $E(\hat{B}^i(t)) \subset \bigcup_{m\neq i} E(\hat{B}^m(t))$ where $\hat{B}^m(t) \subset C^i(t)$, then the expansion will be performed and the blocking of $B^i(t)$ occurs.

Fig. 4 illustrates the case where we avoid blocking event of B^k. B^i is not expanded even though v^i has moved along the arrows into the shaded area. Since the shaded area is not occupied by B^i, enclosure inside B^k can expand to the shaded area not overlapping the enclosures for other vehicles. This prevents the occurrence of blocking event depicted in Fig. 3.

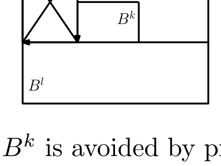

Fig. 4. Blocking of B^k is avoided by preventing expansion of B^i into the shaded region.

The procedure in avoiding blocking only works for the connected graph $C^i(t)$. There exist situations where com-

Table 1. Table of Data Structures and Operations

$\hat{B}^i$: circularly linked list representing the current enclosing boundary for v^i.
$\hat{B}^i.seg(head, tail)$: segment of $\hat{B}^i$ that starts from the head and ends at the tail.
$L_r = \hat{B}^i.Remove(L_s)$: remove linked list L_s from $\hat{B}^i$ resulting in L_r.
CS : singly linked list representing candidate segment.
$\hat{B}^i_u = L_r.Combine(CS)$: combine linked list L_r with CS resulting in updated enclosing boundary $\hat{B}^i_u$.
$I_o = L_s.Search(unexplored)$: search for an unexplored intersection in the linked list L_s and mark the unexplored intersection as I_o. If there is no unexplored intersection, return $NULL$.
D^i : disabled intersection set for v^i.
D^i_k : disabled intersection set of B^i_k.
Set.Store($Data$) : store $Data$ in Set.
$C^i.Search(Q_{v^i})$: search for Q_{v^i} among the unexplored intersections on C^i.
$C^i.Broadcast(Data)$: broadcast $Data$ using C^i to every vehicle sharing C^i.
$C^i.Receive(Data)$: update C^i after receiving $Data$ from every vehicle sharing C^i.

munication graphs $C^i(t)$ and $C^l(t)$ are not connected. In this case, blocking can not be detected. In Fig.5, we illustrate this case where $C^l(t)$ is not connected to $C^i(t)$. Hence, existence of $C^l(t)$ is unknown to the vehicles that are only aware of $C^i(t)$. In this situation, the expansion of $\hat{B}^l(t)$ will eventually be blocked.

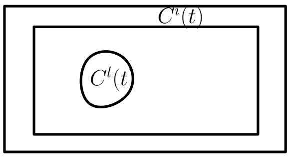

Fig. 5. $C^l(t)$ is not connected to $C^i(t)$. Hence, existence of $C^l(t)$ is unknown to the vehicles that are only aware of $C^i(t)$.

The SCENT algorithms are summarized by Algorithms 1 and 2. and the following table lists the data structure used in the algorithms to construct enclosing boundaries, and communication graphs.

4. PERFORMANCE ANALYSIS

In this section, we provide analytical formula for the total time spent on cooperative exploration of a regularized workspace. Each obstacle other than O_M is now simplified as one site (called a generator in Du et al. (1999)), then the workspace is partitioned by a centroidal Voronoi tessellation. In the case where there are sufficiently many Voronoi cells, each can be shown to be of hexagonal shape Du et al. (1999); Newman (1982). We analyze the performance of our algorithms in this workspace where each cell has a hexagonal shape with identical size, as shown in Fig. 6 and Fig. 7.

We acknowledge that the workspace with hexagonal Voronoi cells is not a realistic configuration in most environments. However, we can achieve hexagonal Voronoi cells using identical circular obstacles. To obtain hexagonal Voronoi cells, all the obstacles satisfy the following two conditions :

Algorithm 1 Construct the Initial Enclosing Boundary for v^i

$n \leftarrow 1$.
$AreaSum \leftarrow 0$.
repeat
  v^i encounters an intersection. Update G^i and C^i.
  $C^i.Receive(G^j, \hat{B}^j)$ for all $G^j \subset C^i$.
  $C^i.Broadcast(G^i)$.
  $E^i_{n,0} \leftarrow$ the intersection.
  Search for an open sector in the counter clockwise direction from sector 0. v^i moves through the first open sector. $E^i_{n,0}.pointer \leftarrow$ first open sector.
  if $E^i_{n,0}$ has an open sector that has not been visited by vehicles **then**
    $E^i_{n,0}.mark \leftarrow unexplored.$
  else
    $E^i_{n,0}.mark \leftarrow explored.$
  end if
  $\hat{B}^i.Insert(E^i_{n,0})$.
  $n \leftarrow n + 1$.
until v^i encounters the $E^i_{1,0}$ for the second time.
if $N(\hat{B}^i) \subset N(\hat{B}^j), j \neq i$ **then**
  $C^i.Search(Q_{v^i})$.
  $C^i.Broadcast(Q_{v^i})$. v^i moves along the shortest path for reaching the Q_{v^i}. Repeat Algorithm 1.
else
  $C^i.Broadcast(\hat{B}^i)$.
  $AreaSum = AreaSum + AreaInside\hat{B}^i$. Implement Algorithm 2.
end if

(1) each obstacle, other than O_M, has identical circular shape.
(2) if we increase the radius of any obstacle other than O_M, the obstacle becomes tangential to nearby obstacles at six points. And these six points correspond to the vertices of hexagon.

Fig.6 illustrates one hexagonal Voronoi cell in the workspace. To make a hexagonal Voronoi cell, one obstacle has a circular shape and O_M has a symmetric shape.

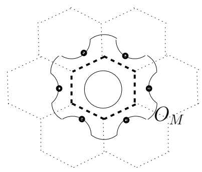

Fig. 6. To make a hexagonal Voronoi cell, one obstacle has a circular shape and O_M has a symmetric shape. If we increase the radius of this circular obstacle, the obstacle becomes tangential to O_M at six points. These six points are marked on O_M.

4.1 Time upper bound of algorithms using one vehicle

We first present result for time upper bound for one vehicle to explore the entire bounded workspace.

Theorem 3. Consider workspace W and one vehicle satisfying assumptions (A1)-(A4). There are M obstacles in W. Except for O_M, every obstacle has hexagonal Voronoi cell with identical size. One vehicle explores W using

Algorithm 2 Expand the Enclosing Boundary for v^i

$A_w \leftarrow$ area inside $\partial V(O_M)$. N is the number of intersections on $\hat{B}^i$. Label the intersections on $\hat{B}^i$ in the clockwise direction as $E^i_{1,0}$,$E^i_{2,0}$,...,$E^i_{N,0}$. $n \leftarrow 1$. $k \leftarrow 0$.
while $AreaSum < \frac{A_w}{N_v}$ and there exists an unexplored intersection on C^i **do**
  v^i visits $E^i_{n,k}$ on $\hat{B}^i$. $C^i.Receive(G^j, \hat{B}^j)$,$\forall G^j \subset C^i$.
  if $E^i_{n,k} \notin \{D^i, D^i_k\}$, and there exists open sector outside $\hat{B}^i$, that satisfies the sector selection rule **then**
    $m \leftarrow 1$. $S_1 \leftarrow E^i_{n,k}$.
    while 1 **do**
      v^i finds S_m. Update G^i. $C^i.Broadcast(G^i)$. Move through the sector selected using the sector selection rule. $S_m.pointer \leftarrow$ selected sector.
      if S_m has an open sector that has not been visited by vehicles **then**
        $S_m.mark \leftarrow unexplored$.
      else
        $S_m.mark \leftarrow explored$.
      end if
      $CS.Insert(S_m)$.
      if $m \neq 1$ and $S_m == E^i_{t,k}$ for any t **then**
        break;
      else
        $m \leftarrow m + 1$.
      end if
    end while
    $head \leftarrow S_1$. $tail \leftarrow E^i_{t,k}$. $HT = \hat{B}^i.seg(head, tail)$.
    if $HT \neq NULL$ and $(HT.Search(unexplored) \in \{NULL, head, tail, (head, tail)\}$ **then**
      $L_r = \hat{B}^i.Remove(HT)$.
      $\hat{B}^i_u = L_r.Combine(CS)$.
      if updated enclosing boundary $\hat{B}^i_u$ leads to $E(\hat{B}^k(t)) \subset \bigcup_{m\neq k} E(\hat{B}^m(t)) \bigcup E(\hat{B}^i_u)$ where $k \neq i$ and $\hat{B}^k(t), \hat{B}^m(t) \subset C^i(t)$ **then**
        $n \leftarrow t + 1$.
      else
        $\hat{B}^i \leftarrow \hat{B}^i_u$. $C^i.Broadcast(\hat{B}^i)$. N is the number of intersections on $\hat{B}^i$. $E^i_{1,k+1} \leftarrow tail$. Relabel the intersections on $\hat{B}^i$ in the clockwise direction as $E^i_{1,k+1}$,$E^i_{2,k+1}$,...,$E^i_{N,k+1}$. $n \leftarrow 1$. $k \leftarrow k + 1$.
      end if
    else
      D^i_k.Store($head$). $n \leftarrow t + 1$.
    end if
  else
    $n \leftarrow n + 1$.
  end if
  if $(E(\hat{B}^i) \subset \bigcup_{m\neq i} E(\hat{B}^m)$ where $\hat{B}^m(t) \subset C^i(t))$ **then**
    $C^i.Search(Q_{v^i})$. $C^i.Broadcast(Q_{v^i})$. v^i moves along the shortest path for reaching the Q_{v^i}. Repeat Algorithm 1 with $AreaSum = AreaSum + AreaInside\hat{B}^i$ not initializing $AreaSum$.
  end if
  if v^i receives $Q_{v^l} (l \neq i)$ from v^l **then**
    D^i.Store(Q_{v^l}).
  end if
end while

the boundary expansion algorithms in Kim et al. (2009). Then, the time for the exploration is bounded above as $T_c < T(\frac{5}{2}(M-1)^2 - \frac{5}{2}(M-1) + 1)$ where T denotes time interval for a vehicle to move along one hexagonal Voronoi Cell.

Proof. Recall that B_k denotes the enclosing boundary for one vehicle updated after k steps and that $k+1$ obstacles are inside B_k. Since B_k is a simple closed curve, B_k divides hexagonal Voronoi cells into two groups : k+1 Voronoi cells inside B_k, and Voronoi cells outside B_k.

Consider the case where $k = 0$. Since there is only one Voronoi cell inside B_0, time interval to construct B_0 is

$$T_{B_0} = T, \tag{1}$$

where T denotes time interval for a vehicle to move along one hexagonal Voronoi Cell. Next, consider the case where $k > 0$, i.e., there are more than one Voronoi cell inside B_k. In this case, any Voronoi cell inside B_k is adjacent to at least one other Voronoi cell inside B_k as illustrated on Fig. 7. Since there are at most 6 adjacent Voronoi cells for every hexagonal Voronoi cell, there exist at most 5 adjacent Voronoi cells outside B_k for any Voronoi cell inside B_k. Thus, for $k + 1$ Voronoi cells inside B_k, we can have at most $5(k + 1)$ adjacent Voronoi cells outside B_k, i.e., upper bound for the number of Voronoi cells intersecting perimeter of B_k is $5(k + 1)$.

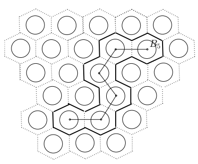

Fig. 7. All Voronoi cells, except for $V(O_M)$, have hexagonal shapes with identical size. Line segment connecting two centers of Voronoi cells inside B_k represents the adjacency of two Voronoi cells inside B_k.

We update the enclosing boundary until all obstacles except for O_M are inside the enclosing boundary. After B_{k+1} is generated, one addable obstacle is inside the enclosing boundary using Theorem 2 in Kim et al. (2009). This implies that one of the Voronoi cells, outside B_k, intersecting perimeter of B_k is inside B_{k+1}. Therefore, we have

$$T_{B_{k+1}} < T_{B_k} + (5k + 5)T, \tag{2}$$

where T_{B_k} denotes the time interval for a vehicle to construct B_k. Using (2), we obtain

$$\begin{aligned} T_{B_k} &< 5(1 + 2 + ... + (k-1))T + 5kT + T_{B_0} \\ &= T(\frac{5}{2}k^2 + \frac{5}{2}k + 1), \end{aligned} \tag{3}$$

since $T_{B_0} = T$ using (1). Recall that there are $k + 1$ obstacles inside B_k. Also, inside the area enclosed by $\partial V(O_M)$, there are $M-1$ obstacles with hexagonal Voronoi cells. Therefore, our algorithm terminates when

$$k + 1 = M - 1. \tag{4}$$

Furthermore, using (3), time upper bound for the construction of Voronoi diagram is

$$T_c < T(\frac{5}{2}(M-1)^2 - \frac{5}{2}(M-1) + 1). \tag{5}$$

Therefore, expected construction time is $O((M-1)^2)$. $\square$

4.2 Time upper bound of algorithms using multiple vehicles

For multiple vehicles, SCENT algorithms end when the area inside the enclosing boundary built by v^i is greater than $\frac{A_w}{N_v}$ where A_w is the total area of the workspace. Upper bound for the exploration time is claimed by Theorem 4.

Theorem 4. Consider workspace W and vehicles satisfying assumptions (A1)-(A5). In W, there are M obstacles. Except for O_M, every obstacle has hexagonal Voronoi cell with identical size. Suppose that there exist N_v vehicles and that time upper bound for a vehicle's traversal along perimeter of W is T_o. Every vehicle explores W using SCENT algorithms. Then total time spent to construct a complete Voronoi diagram is bounded above by $T + \lceil\frac{M-1}{N_v}\rceil(T_o+T)+\frac{5}{2}T\lceil\frac{M-1}{N_v}\rceil+\frac{5}{2}T(\lceil\frac{M-1}{N_v}\rceil)^2$ where T denotes time interval for a vehicle to move along one hexagonal Voronoi Cell.

Proof.

SCENT algorithms for v^i end when the area inside the enclosing boundary built by v^i is greater than $\frac{A_w}{N_v}$. Since $\frac{A_w}{N_v}N_v = A_w$, total time spent to construct a complete Voronoi diagram is time upper bound of SCENT algorithms for one vehicle.

Suppose that blocking occurs $l-1$ times before algorithms terminate. Since number of obstacles is finite, l is also finite. Let k_i denote the updated step of enclosing boundary before each blocking occurs. In other words, after the enclosing boundary is updated after k_i steps, blocking occurs and a new enclosing boundary is built at new position. Then the enclosing boundary is updated after k_{i+1} steps before blocking occurs again. In this way, boundary update occurs in the order of $k_1 \to ... \to k_l$ and blocking occurs between $k_i \to k_{i+1}$ where $1 \le i \le l-1$. Similarly to (4), algorithms finish when

$$\sum_{j=1}^{l}(k_j+1) = \lceil\frac{M-1}{N_v}\rceil. \quad (6)$$

In (6), ceiling function is used since k_j is integer. Furthermore, we obtain

$$\sum_{j=1}^{l}k_j = \lceil\frac{M-1}{N_v}\rceil - l \ge 0, \quad (7)$$

where inequality holds, since $k_j \ge 0$ for all j.

We suppose that boundary update occurs in the order of $k_1 \to ... \to k_l$ and that blocking occurs between $k_i \to k_{i+1}$. Hence, blocking occurs $l-1$ times. Once blocking of B^i happens, v^i moves along the shortest path on $C_i(t)$ to reach Q_{v^i}. Since v^i moves along the shortest path on $C_i(t)$ to reach Q_{v^i}, time spent to reach Q_{v^i} is upper bounded by T_o. Recall that T_o is time upper bound for a vehicle's traversal along perimeter of W. Next, we get the construction time bound as

$$T_c \le T_{B_0} + T_o + \sum_{j=1}^{l}T(\frac{5}{2}k_j^2+\frac{5}{2}k_j+1)+(l-1)T_o, \quad (8)$$

where (3) is used. Note that $T_{B_0}+T_o$ is added in front of right side of (8) considering the overlapping of B^i. Here, T_{B_0} is the time interval to build B_0^i, since overlapping of B^i $(N(\hat{B}^i(t)) \subset N(\hat{B}^j(t))(j \ne i))$ is detectable after B_0^i is built. Recall that, once overlapping of B^i is detected, v^i moves along the shortest path on $C_i(t)$ to reach the Q_{v^i}. Since v^i moves along the shortest path on $C_i(t)$ to reach the Q_{v^i}, time spent to reach Q_{v^i} is upper bounded by T_o. Now, we express (8) as a function of N_v and $M-1$. Using (7) and (8), we obtain

$$\begin{aligned} T_c &< T_{B_0} + lT_o + lT + \frac{5}{2}T(\lceil\frac{M-1}{N_v}\rceil - l) \\ &+\frac{5}{2}T(\lceil\frac{M-1}{N_v}\rceil - l)^2, \end{aligned} \quad (9)$$

since $\sum_{i=1}^{l}(k_i^2) \le (\sum_{i=1}^{l}k_i)^2$. Furthermore, using (7) and (1), we get

$$\begin{aligned} T_c &< T + \lceil\frac{M-1}{N_v}\rceil(T_o+T) + \frac{5}{2}T\lceil\frac{M-1}{N_v}\rceil \\ &+\frac{5}{2}T(\lceil\frac{M-1}{N_v}\rceil)^2. \end{aligned} \quad (10)$$

$\square$

In the case where $\frac{M-1}{N_v} \gg 1$, (10) is approximated as

$$T_c < \frac{M-1}{N_v}(T_o(M)) + \frac{5}{2}T(\frac{M-1}{N_v})^2, \quad (11)$$

where $T_o(M)$ is used to indicate the fact that T_o is a function of M. (11) implies that as N_v increases by n times, the maximum exploration time decreases by n times.

Suppose that $\frac{M-1}{N_v} \gg 1$ and that overlapping or blocking is completely avoided. Then no vehicle has to be redirected to build a new enclosing boundary at new position. In this case, from (11), we obtain

$$T_c < \frac{5}{2}T(\frac{M-1}{N_v})^2, \quad (12)$$

since $T_o(M)$ term in (11) is related to the time interval for building a new enclosing boundary at new position. (12) implies that as N_v increases by n times, upper bound of exploration time decreases by n^2 times.

Next, consider the case where N_v increases to ∞ ($\lceil\frac{M-1}{N_v}\rceil \to 1$). Then, from (10), we obtain

$$T_c < T_o(M) + 7T, \quad (13)$$

which implies that, as the number of vehicle increases, the effectiveness of adding more vehicles decreases. However, we acknowledge that, as N_v increases, assumption (A5) gets more difficult to be satisfied.

5. SIMULATION RESULTS

We compare the boundary expansion algorithms for one vehicle with the SCENT algorithms using two vehicles in MATLAB.

Fig. 8 shows one vehicle constructing a Voronoi diagram in a rectangular shaped workspace. At every intersection, the vehicle decides the heading direction based on the boundary expansion algorithms. Then the control law

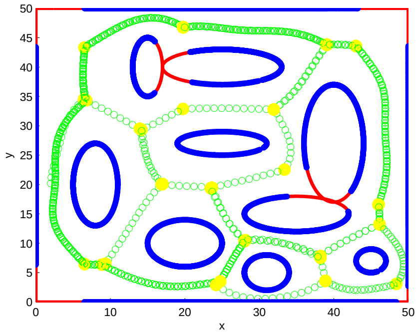

Fig. 8. Trajectory of one vehicle that constructs a Voronoi diagram in an unknown area.

(Kim et al. (2009)) is applied to make a vehicle move along the edges.

The obstacle boundary and the points on the obstacle boundary, detected using the range scanner, are shown as red and blue lines respectively. The trajectory of a vehicle is presented with green line. Also, on the vehicle's trajectory, intersections are presented as large yellow dots. We observe that all the intersections are completely explored by the vehicle. Time spent to construct the Voronoi diagram is 99.05 time unit.

Fig. 9 shows two vehicles constructing a Voronoi diagram in a rectangular shaped workspace using the SCENT algorithms. The obstacle boundary and the points on the obstacle boundary, detected using the range scanner, are shown as red and blue lines respectively. Initial positions of two vehicle are $(2, 20)$ and $(45, 2)$ respectively. The trajectory of two vehicles are marked with green circle and black point respectively. We observe that all the intersections are completely explored by two vehicles. Exploration time using two vehicles is 29.67 time unit which is less than one third of exploration time using one vehicle.

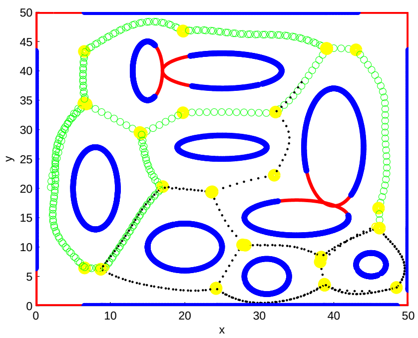

Fig. 9. Trajectory of two vehicles that constructs a Voronoi diagram in an unknown area.

6. CONCLUSION

We develop the SCENT algorithms based on the RGVG. Exploration algorithm of a single vehicle is used for each vehicle until the information network intersects. A communication graph based on the RGVG can be utilized to detect blocking or overlapping between vehicles. These blocked or overlapped vehicles are then redirected to unexplored regions of the workspace using the communication graph. Moreover, the communication graph is applied to reduce the chance of occurrence of blocking events for multiple vehicles. We prove that such a cooperative strategy leads to time efficient construction of the RGVG in an unknown workspace.

REFERENCES

Aurenhammer, F. (1991). Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23, 345–405.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and map making: Part II. *IEEE Robotics and Automation Magzine*, 108–117.

Choset, H. and Burdick, J. (2000). Sensor-based exploration: The hierarchical generalized voronoi diagram. *The International Journal of Robotics Research*, 19, 96–125.

Choset, H., Konukseven, I., and Burdick, J. (1996). Mobile robot navigation: issues in implementating the generalizedvoronoi graph in the plane. In *Proc. of IEEE/SICE/RSJ International Conference on Multi-sensor Fusion and Integration for Intelligent Systems*, 241 – 248. Washington DC.

Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2), 243–255.

Culler, D., Estrin, D., and Srivastava, M. (2004). Overview of sensor networks. *IEEE Computer Magazine*, 37(8), 41–49.

Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal voronoi tessellations : Applications and algorithms. *Society for Industrial and Applied Mathematics*, 41, 637–676.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and map making: Part I. *IEEE Robotics and Automation Magzine*, 99–108.

Ferrari-Trecate, G., Egerstedt, M., Buffa, A., and Ji, M. (2006). Laplacian sheep: A hybrid, stop-go policy for leader-based containment control. *Hybrid Systems: Computation and Control*, 3927, 212–226.

Kim, J., Zhang, F., and Egerstedt, M. (2009). An exploration strategy based on construction of voronoi diagrams. In *IEEE Conference on decision and control*. Shanghai, China.

Klein, R. (1990). *Concrete and Abstract Voronoi diagrams*. Springer.

Lavalle, S.M. (2006). *PLANNING ALGORITHMS*. Cambridge University Press.

Martinez, S., Cortes, J., and Bullo, F. (2007). Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27(4), 75 – 88.

Nagatani, K. and Choset, H. (1999). Toward robust sensor based exploration by constructing reduced generalized voronoi graph. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1687–1692.

Newman, D. (1982). The hexagon theorem. *IEEE Transactions on Information Theory*, 28, 137–139.

Thurn, S. and Burgard, W. (2005). *Probabilistic Robotics*. MIT, Cambridge, MA.